IN THE CLAIMS:

1. (Canceled)

2. (Currently Amended) The method as in claim ~~3~~ 53 further including populating a destination image with extracted contents of the source disk in which the destination image has files, attributes, and structural relationships between files identical to files associated with the source disk.

3. (Canceled)

4. (Currently Amended) The method as in claim ~~3~~ 53, further comprising forwarding the intercepted sector-based I/O requests to the first computer over a network.

5. (Currently Amended) The method as in claim ~~3~~ 53, in which the source disk is associated with a first computer that has a memory, further comprising:

loading an imaging client program in the memory of the first computer, the imaging client program not being resident on the source disk; and

passing the intercepted sector-based I/O requests to the imaging client program, the imaging client program directing the intercepted sector-based I/O requests to the source disk.

6. (Currently Amended) The method as in claim ~~3~~ 53, further comprising:

loading a secondary operating system in the memory of the first computer, said secondary operating system not being present on the source disk and mediating I/O requests between the imaging client program and the source disk.

7. (Previously Presented) The method as in claim 2 further comprising:

mounting the destination image in an uninitialized state in the second computer as a simulated destination disk;

intercepting sector-based I/O requests directed to the simulated destination disk and directing the contents of the intercepted sector-based I/O requests to the destination image;

retrieving partition and file system layout information from the source disk;

formatting the simulated destination image to have the same partitioning and file system as the simulated source disk and thus of the source disk; and

copying files of at least one file system of the simulated source disk to the corresponding file system of the simulated destination disk.

8. (Previously Presented) The method as in claim 7, further comprising converting the intercepted sector-based I/O requests to the simulated destination disk into sector accesses within the destination image.

9. (Previously Presented) The method as in claim 7, in which the destination image is a virtual disk file associated with a virtual computer.

10. (Previously Presented) The method as in claim 9, in which the first computer is a physical computer and the source disk is a physical disk associated with the physical computer.

11. (Previously Presented) The method as in claim 9, in which the virtual disk file is a sparse virtual disk, having a predetermined capacity and initial sector contents with null values.

12. (Previously Presented) The method as in claim 7, in which the source disk is a source virtual disk.

13. (Previously Presented) The method as in claim 12, in which the destination disk is a physical disk.

14. (Previously Presented) The method as in claim 7, in which the source disk is a first virtual disk associated with a first virtual computer and the destination disk is a second virtual disk associated with a second virtual computer.

15. (Previously Presented) The method as in claim 7, in which the first computer is the same as the second computer.

16 -18. (Canceled)

19. (Currently Amended) The system as in claim ~~18~~ 58, further comprising a network adapter, residing in said memory, to forward the intercepted sector-based I/O requests to the first computer.

20. (Previously Presented) The system as in claim 19, further comprising:

a first computer memory within the first computer; and

an imaging client installed in the first computer memory, said imaging client comprising computer-executable instructions that include code to receive any source disk I/O requests issued from the second computer to the first computer, code to direct the intercepted sector-based I/O requests to the source disk, and code to pass the retrieved source disk data to the second computer in response to the source disk I/O requests.

21. (Currently Amended) The system as in claim ~~18~~ 58 wherein the imaging server further includes code to generate a simulated destination disk in response to the second computer mounting the destination image, with said memory further including a local loop-back driver, a local adapter and a formatting module, with the local loop-back driver intercepting sector-based I/O requests directed to the simulated destination disk and retrieving partition and file system layout information from the source disk, the local adapter comprising code to convert the intercepted sector-based I/O requests to the simulated destination disk into sector accesses within the destination image and the formatting module comprising code to format the destination image to have the same partitioning and file system(s) as the simulated source disk and thus of the source disk, the imaging server having code to copy files of at least one file system of the simulated source disk to the corresponding file system of the simulated destination disk.

22. (Previously Presented) The system as in claim 21, in which the source disk is a virtual disk.

23. (Previously Presented) The system as in claim 22, in which the destination disk is a physical disk.

24. (Previously Presented) The system as in claim 21, in which the destination image is a virtual disk file associated with a virtual computer.

25. (Previously Presented) The system as in claim 24, in which the first computer is a physical computer and the source disk is a physical disk associated with the physical computer.

26-52. (Canceled)

53. (New) A method for creating an image of a source disk of a first computer on a destination disk of a second computer that includes an operating system and file system software, said method comprising:

mounting in the second computer, a simulated source disk corresponding to the source disk of the first computer such that the simulated source disk is accessible by the operating system in the second computer as a local disk, the file system software within the second computer detecting a file system of the mounted simulated source disk and therefore of the source disk, exposing the file system to the software running on the second computer, and issuing sector-based I/O requests toward the simulated source disk;

intercepting sector-based I/O requests directed to the simulated source disk and retrieving source disk data from the source disk according to the intercepted sector-based I/O requests such that contents of the source disk in the first computer are extracted at the sector level and system software in the first computer need not detect the file system of the source disk; and

populating a destination image in the destination disk of the second computer with the contents of the source disk such that the destination image has a different sector-by-sector content than the source disk but a destination file system logically equivalent to the source file system.

54. (New) A method as in claim 53, wherein the simulated source disk is a file presented to the operating system of the second computer as a physical disk.

55. (New) A method as in claim 53, wherein the source disk in the first computer need not be modified prior to mounting the simulated source disk in the second computer.

56. (New) A method for creating an image of a source disk of a first computer on a destination disk of a second computer that includes an operating system and file system software, said method comprising:

mounting in the second computer, a simulated source disk corresponding to the source disk of the first computer such that the simulated source disk is accessible by the operating

system in the second computer as a local disk, the file system software within the second computer detecting a file system of the mounted simulated source disk and therefore of the source disk, exposing the file system to the software running on the second computer, and
issuing sector-based I/O requests toward the simulated source disk;

intercepting sector-based I/O requests directed to the simulated source disk and retrieving source disk data from the source disk according to the intercepted sector-based I/O requests such that contents of the source disk in the first computer are extracted at the sector level and system software in the first computer need not detect the file system of the source disk;

creating a destination image as a virtual disk;

mounting the destination image in an uninitialized state in the second computer as a simulated destination disk accessible by the operating system of the second computer, such that sector-based I/O requests directed to the simulated destination disk are intercepted and converted into sector accesses within the destination image;

retrieving partition and file system layout information from the source disk in the first computer;

formatting the simulated destination image to have the same partitioning and file system(s) as the simulated source disk, which has the same partitioning and file system(s) as the source disk in the first computer; and

populating the destination image in the destination disk of the second computer with the contents of the source disk such that the destination image has a different sector-by-sector content than the source disk but a destination file system logically equivalent to the source file system.

57. (New) The method of claim 56 further comprising:

loading an imaging client program into a memory of the first computer;

passing the intercepted sector-based I/O requests to the imaging client program, the imaging client program directing the intercepted sector-based I/O requests to the source disk; and

mediating, by the operating system, sector-based I/O requests between the imaging client program and the source disk.

58. (New) A system for creating an image of a source disk of a first computer with contents arranged according to at least one file system on a destination disk of a second computer, said system comprising:

a first computer having the source disk; and

a second computer having a memory with an operating system, file system software, and a destination disk and including computer executable instructions having code:

for creating a simulated source disk with a representation of information stored on the source disk in the first computer;

for mounting the simulated source disk in the second computer such that the simulated source disk is accessible by the operating system in the second computer as a local disk, the memory of the first computer further including file system software to detect a file system of the simulated source disk, and therefore of the source disk, to expose the file system to the software running on the second computer, and to issue sector-based I/O requests toward the simulated source disk;

for intercepting sector-based I/O requests directed to the simulated source disk and retrieving source disk data from the source disk according to the intercepted sector-based I/O requests such that contents of the source disk in the first computer are extracted at the sector level and system software in the first computer need not detect the file system of the source disk; and

for populating a destination image in the destination disk of the second computer with the contents of the source disk such that the destination image has a different sector-by-sector content than the source disk but a destination file system logically equivalent to the source file system.

59. A system for creating an image of a source disk of a first computer, which has a memory and in which contents of the source disk are arranged according to at least one source file system, comprising:

a second computer including an operating system, file system software, and a processor and memory coupled to the processor configured to provide the processor with instructions for:

mounting in the second computer, a simulated source disk corresponding to the source disk of the first computer such that the simulated source disk is accessible by the operating system in the second computer as a local disk, the file system software within the second computer detecting a file system of the mounted simulated source disk and therefore of the source disk, exposing the file system to the software running on the second computer, and
issuing sector-based I/O requests toward the simulated source disk;

intercepting sector-based I/O requests directed to the simulated source disk and retrieving source disk data from the source disk according to the intercepted sector-based I/O requests such that contents of the source disk in the first computer are extracted at the sector level and system software in the first computer need not detect the file system of the source disk;

creating a destination image as a virtual disk;

mounting the destination image in an uninitialized state in the second computer as a simulated destination disk accessible by the operating system of the second computer, such that sector-based I/O requests directed to the simulated destination disk are intercepted and converted into sector accesses within the destination image;

retrieving partition and file system layout information from the source disk in the first computer;

formatting the simulated destination image to have the same partitioning and file system(s) as the simulated source disk, which has the same partitioning and file system(s) as the source disk in the first computer; and

populating the destination image in the destination disk of the second computer with the contents of the source disk such that the destination image has a different sector-by-sector content than the source disk but a destination file system logically equivalent to the source file system.

60. (New) The system of claim 59 further comprising

an imaging client installed in the memory of the first computer, said imaging client comprising computer-executable instructions:

for receiving any source disk I/O requests issued from the second computer to the first computer,

for directing the intercepted sector-based I/O requests to the source disk, and

for passing to the second computer source disk data retrieved in response to the source disk I/O requests.